The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Active sampling based on Gaussian process for reinforcement learning

Kazuhiro Takeda, Takeshi Mori
Guraduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara, 630-0192, Japan

Shin Ishii
Guraduate School of Informatics
Kyoto University
Gokasho, Uji, Kyoto, 611-0011, Japan

Abstract

In reinforcement learning (RL), many samples are necessary in every policy improvement, which requires the robot actually to act many times and hence may make the robot be broken down. One cause of the necessity of many samples in the RL is that the agent must passively produce samples according to it's current policy. Therefore, efficient sampling methods such as active sampling are desired. In this article, we propose a novel RL method with active sampling based on Gaussian process, which reduces samples necessary for policy convergence.

1 Introduction

Reinforcement learning (RL) is a machine learning technique which seeks good policy through interactions with environments, and has been applied to various control problems. Recently, policy gradient RL methods have been proposed, in which the gradient of the cumulative rewards with respect to a the parameter of the policy (policy gradient) is estimated from samples and the policy is improved by using the gradient [7]. In this method, the convergence of the policy parameter is guaranteed and more robust learning can be expected than the conventional RL methods do through many experimental studies. In the RL algorithms including the policy gradient methods, however, many samples are necessary in every policy improvement, which requires the robot actually to act many times and hence may make the robot be broken down. This is a main drawback of the RL algorithms in many real-world applications.

One cause of the necessity of many samples is that the agent must passively produce samples according to it's current policy. The passive sampling needs many samples for achieving successful learning, because less efficient samples are produced frequently. Therefore, efficient sampling methods such as active sampling are desired.

One possible choice in this direction can be provided by Bayesian approaches, where the estimation variance of the policy gradient or the value function can be estimated from samples. Then, we can know which samples reduce variance of the policy gradient with estimation of the variance in an online manner. If we draw samples which efficiently reduce the estimation variance, the policy gradient can be estimated based on a fewer number of samples than that in the conventional passively sampling RL methods; then, efficient policy improvement can be done.

An application of the nonparametric Bayesian approach such as Gaussian processes (GPs) to the RL is rather straightforward for this end, because the posterior distribution of arbitrary function can be estimated without any prior knowledge. Recently, a policy gradient RL method which is called "Bayesian Policy Gradient Algorithm (BPG)" was proposed [1]. In this method, the GP is combined with the conventional policy gradient methods and the estimation variance of the policy gradient is derived. However, there is no study of the active sampling by using such an estimation variance as a sampling criterion.

In this article, we propose a novel RL method with active sampling using the estimation variance as a sampling criterion, called "Off-Policy Bayesian Policy Gradient Algorithm (Off-BPG)". In our proposal method, an active sampling is performed toward reducing the estimation variance in an online manner, instead of the conventional passive sampling. Then, the policy improvements can be faster and may be more plausible to applications to real-world problems with scarce samples. Computer experiments show that the policy gradient could be estimated efficiently with fewer sampling times by our method rather than by the previous BPG.

2 MDPs and Policy Gradient algorithm

We consider discrete-time and continuous-state Markov decision processes (MDPs). Let the state and the action be s_t and a_t at time t, respectively. The agent selects the action a_t from the probabilistic policy $(a_t|s_t;\boldsymbol{\theta})$, prescribed by the parameter $\boldsymbol{\theta}$, at the state s_t, and moves to the next state s_{t+1} according to the state transtion distribution $p(s_{t+1}|s_t,a_t)$, and recieves a reward r_{t+1}. We define the sample trajectory generated by this Markov chain as a single path by $x = (s_0, a_0, s_1, a_1, \cdots, s_{T-1}, a_{T-1}, s_T)$. The distribution of generating such a path is given by

$$p(x;\boldsymbol{\theta}) = p_0(s_0)\prod_{t=0}^{T-1} \quad (a_t|s_t;\boldsymbol{\theta})p(s_{t+1}|s_t,a_t).$$

The cumulative reward of the path x is defined as $R(x) = \sum_{t=0}^{T-1} \ ^t r_{t+1}$, where $\ \in [0,1)$ is a discount factor. The expected value of $R(x)$ for a given path x is defined as $R(x)$. Then, the expected return of cumulative reward under policy is given by

$$\eta(\boldsymbol{\theta}) = \eta(\ (\cdot|\cdot;\boldsymbol{\theta})) = \int R(x)p(x;\boldsymbol{\theta})dx.$$

In the policy gradient methods, the derivative of the expected return with respect to the policy parameters $\boldsymbol{\theta}$ (policy gradient) is estimated, and then the policy is improved by updating the parameters in the direction of the gradient. The policy gradient is calculated as

$$\begin{aligned}\nabla\eta(\boldsymbol{\theta}) &= \int R(x)\nabla\log p(x;\boldsymbol{\theta})p(x;\boldsymbol{\theta})dx \\ &= \int R(x)\sum_{t=0}^{T-1}\nabla\log \ (a_t|s_t;\boldsymbol{\theta})p(x;\boldsymbol{\theta})dx.\end{aligned}$$

3 Active Sampling based on Gaussian Process

Although the policy gradient methods perform robust learning in many experimental studies, many samples are necessary due to large variance of the gradient estimation. Some efficient sampling methods such as active sampling are desired for variance reduction and hence fast learning. In this section, we introduce two types of variance reduction techniques based on the GP. The GP [3] is a non-parametric Bayesian technique for modeling real-world statistical problems without explicit representations of the basis functions. Then, the GP would be promising in the RL field, because the basis function construction in the RL is often very difficult.

3.1 Active Sampling for function $f(x)$

According to GP, we can estimate the posterior distribution of an arbitrary function by setting a Gaussian distribution as a prior of the target function, and observing a number of data from the function. We set the prior mean and the prior variance of the function $f(x)$ to 0 and $k(x,x')$, respectively, where$k(x,x')$ is a kernel function whose inputs are x and x', which represents the similarity between x and x'. We also observe a set of samples $\mathcal{D}_M = \{(x_i,y_i)\}_{i=1}^M$, where x_i and y_i are values of an input and an output of that function, respectively, and M is the number of samples. Then, the posterior distribution is expressed by Gaussian distribution with the posterior mean and the posterior covariance as

$$E_f(f(x)|\mathcal{D}_M) = \boldsymbol{k}_M^\top(x)\boldsymbol{C}_M\boldsymbol{y}_M, \quad (1)$$

$$\begin{aligned}&\mathrm{Cov}_f(f(x),f(x')|\mathcal{D}_M)\\ &\qquad = k(x,x') - \boldsymbol{k}_M^\top(x)\boldsymbol{C}_M\boldsymbol{k}_M(x'), \quad (2)\end{aligned}$$

respectively, where $(^\top)$ is the transpose, and

$$\begin{aligned}\boldsymbol{k}_M(x) &= (k(x_1,x),\ldots,k(x_M,x))^\top,\\ \boldsymbol{y}_M &= (y_1,\ldots,y_M)^\top,\\ \boldsymbol{C}_M &= (\boldsymbol{K}_M+\sigma^2\boldsymbol{I})^{-1},\\ \boldsymbol{K}_{M+1} &= \left[\begin{array}{c|c}\boldsymbol{K}_M & \boldsymbol{k}_M\\ \hline \boldsymbol{k}_M^\top & k(x,x)\end{array}\right].\end{aligned}$$

Here, the vector $\boldsymbol{k}_M(x)$ denotes he similarity between the next input variable x and the current input variables $\{x_1,\ldots,x_M\}$, the matrix $\boldsymbol{K}_M$ denotes the similarity between the current input variables, σ^2 is noise variance of the outputs, which is set in advance, and $\boldsymbol{I}$ is the identity matrix.

Then, we can estimate the posterior variance $\mathrm{Var}_f(f(x)|\mathcal{D}_M) \equiv \mathrm{Cov}_f(f(x),f(x)|\mathcal{D}_M)$ corresponding to the next input variable x, which signifies which input x can largely reduce the variance. The posterior mean (Eq.(1)) and the posterior variance $\mathrm{Var}_f(f(x)|D_M)$ are depicted in Fig.1 (a). After sampling to reduce the posterior variance, the posterior distribution comes to be sharp, and more accurate posterior mean can be obtained, depicted in Fig.1 (b). Therefore, we can estimate the target function $f(x)$ by active sampling based on the GP to reduce the variance largely.

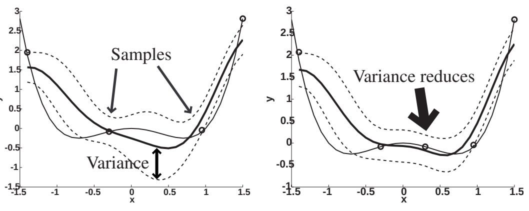


Figure 1: The posterior variance is largely reduced by active sampling. The target function of this problem is $f(x) = x^4 - x^2$. A thick line denotes the posterior mean. A thin line denotes the true function. A dash line denotes the posterior variance. Each circle denotes a sample. Note that in (b), the posterior mean fits the true function more accurately than that in (a), due to the sampling of x as to reduce the posterior variance.

3.2 Active sampling for Expectation $E_x[f(x)]$

If we wish to estimate the expectation of $f(x)$ with respect to x: $E_x[f(x)] = \int f(x)p(x)dx$, the above active sampling method cannot directly be used. In this case, the Bayesian Quadrature (BQ) [9], which is an integral calculus with the GP, can be used for estimating this expectation and its variance. In the BQ, the posterior mean and the posterior variance of the expectation, which we call the BQ mean and the BQ variance, respetively, are given by

$$E_f(\rho|\mathcal{D}_M) = \int E(f(x)|\mathcal{D}_M)p(x)dx, \tag{3}$$

$$\mathrm{Var}_f(\rho|\mathcal{D}_M) = \iint \mathrm{Cov}(f(x), f(x')|\mathcal{D}_M)p(x)p(x')dxdx', \tag{4}$$

where $\rho = E_x[f(x)]$.

Because the new input variable x is marginalized out, we sample x_M shown in Eqs. (1) and (2) to reduce the BQ variance (Eq.(4)). This is another active sampling scheme, which is applied to the BPG in the next section. The effectiveness of this method is depicted in Fig.2, and the algorithm is depicted in Algorithm 1, where samples are obtained by the steepest descent of the BQ variance.

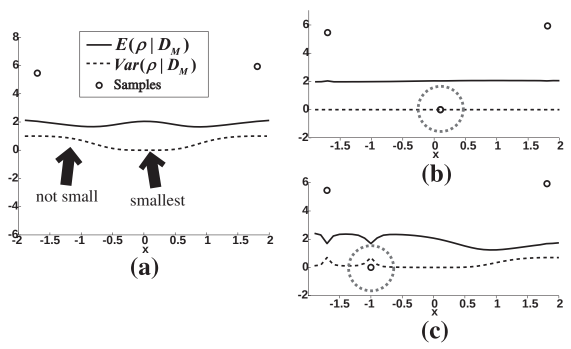


Figure 2: The BQ variance is largely reduced by active sampling, where $\rho = \int(x^4 - x^2)p(x)dx$ and $p(x) = \mathcal{N}(0,1)$. Black straight and dash lines denote the BQ mean and the BQ variance, respectively. Each circle denotes a sample. The objective is to get $\rho = 2$ in all x. (a) shows the BQ mean and the BQ variance in the presence of two samples. (b) shows that we can estimate ρ more accurate with drawing a sample whose BQ variance is small in (a). (c) shows the situation of drawing a sample whose BQ variance is not small, then the estimation of ρ is not more accurate than that in (b).

```
Algorithm 1: Active Sampling based on BQ
1:  input $\mathcal{D}_M$, $p(x)$, learning rate
2:  for h=1 to N   ***steepest descent method***
3:      Compute $\nabla_x\mathrm{Var}(\rho|\mathcal{D}_M)$ using $x_h$
4:      $x_{h+1} = x_h -\ \ \nabla_x\mathrm{Var}(\rho|\mathcal{D}_M)$
5:  end for
6:  return $x$
```

4 Off-Policy Bayesian Policy Gradient

We apply the active sampling method above to the policy gradient RL method. Recently, Bayesian policy gradient algorithm (BPG), employing on the GP, has been proposed by Ghavamzadeh and Engel [1]. In this method, the policy gradient was estimated based on the GP and the BQ. Then, we can apply the active sampling method based on the BQ, proposed in section 3.2, to the BPG. This algorithm is called "Off-policy Bayesian policy gradient algorithm (Off-BPG)". In our Off-BPG and the BPG algorithms, the BQ mean and the BQ covariance of the policy gradient $\nabla\eta(\boldsymbol{\theta})$ are given by [1]:

$$E(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M) = \int E(f(x;\boldsymbol{\theta})|\mathcal{D}_M)p(x;\boldsymbol{\theta})dx,$$

$$\mathrm{Cov}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M) = \iint \mathrm{Cov}(f(x;\boldsymbol{\theta}), f(x';\boldsymbol{\theta})|\mathcal{D}_M)p(x;\boldsymbol{\theta})p(x';\boldsymbol{\theta})dxdx',$$

respectively, which are derived by replacing ρ in Eqs. (3) and (4) with $\nabla\eta(\boldsymbol{\theta})$. Since the BQ covariance is isotropic [1], we define the BQ variance of the policy gradient Var$(\nabla(\boldsymbol{\theta})|\mathcal{D}_M)$ as a diagonal element of $\mathrm{Cov}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M)$. Therefore, we can draw samples efficiently by using $\mathrm{Var}(\nabla\eta(\boldsymbol{\theta})|\mathcal{D}_M)$. This algorithm is depicted in Algorithm 2.

```
Algorithm 2: Off-Policy Bayesian Policy Gradient
1:  input parameterized policy   $(\cdot|\cdot;\boldsymbol{\theta}_0)$, learning rate
2:  for j=1:K   ***policy improvement***
        for i=1:L   ***policy evaluation***
3:          $\mathcal{D}_i$=Algorithm 1$(\mathcal{D}_{i-1},\ \ (\cdot|\cdot;\boldsymbol{\theta}_{j-1})^1)$
4:          Execute $x_{i-1}$ and get $y_{i-1}$
            $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1} + \{x_{i-1}, y_{i-1}\}$
5:          Compute $E(\nabla\eta(\boldsymbol{\theta}_{j-1})|\mathcal{D}_i)$
        end for
6:      $\boldsymbol{\theta}_j = \boldsymbol{\theta}_{j-1} +\ \ E(\nabla\eta(\boldsymbol{\theta}_{j-1})|\mathcal{D}_i)$
7:  end for
    ^1In BPG, we can get $x$ without knowing $p(x)$
      by using the Fisher kernel technique.
```

We collect sample paths whose $\mathrm{Var}(\nabla\rho(\boldsymbol{\theta})|\mathcal{D}_M)$ is small, by using a steepest descent method, and then, the policy gradient is updated.

5 Experimental Results

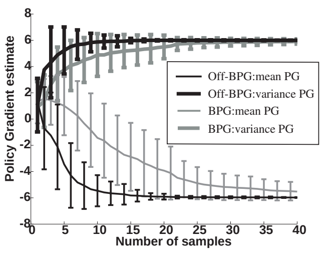
In this section, we compare our off-BPG with the BPG [1] by using a simple bandit problem. First, we evaluate the variance reduction performance of our method in the policy evaluation step. Second, we evaluate the convergence of the policy of our method in the policy improvement step.

5.1 A simple bandit problem

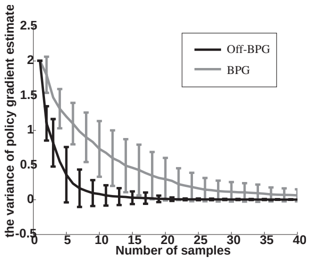
In a simple bandit problem, the path x is equivalent to an action, i.e., $x = a$. So, the policy is reduced to $(a|s) = p(x)$. In this setting, we define the policy as Gaussian distribution, where the mean and the variance are 0 and 3, respectively, namely, $p(x) \equiv \mathcal{N}(0, 3)$. In a usual passive sampling RL setting, when the agent selects an action x from the policy $p(x)$, an immediate reward $r(x)$ is obtained. The reward is set as $r(x) = -(x-3)^2$. On the other hand, we consider the active sampling based on Algorithms 1 and 2, where samples are generated by our active sampling scheme.

5.2 The performance comparsion in the policy evaluation step

We compare our Off-BPG with the BPG in the policy evaluation step. The true gradient is given by $\eta(\boldsymbol{\theta}) = \int -(a-3)^2 \nabla \log p(x) p(x) dx = [6, -6]^\top$. Fig.3 shows the performance comparison, averaged over 100 learning runs. Fig.3 (a) shows that Off-BPG could estimate the policy gradient more efficiently than the BPG, and Fig.3 (b) shows that the variance of Off-BPG decreased faster than the BPG.



(a)

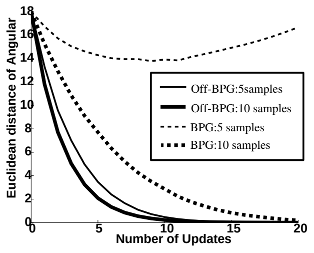


(b)

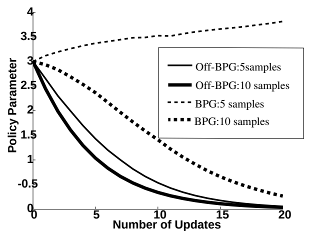
Figure 3: The performance comparison in the policy evaluation step.

5.3 The performance comparsion in the policy improvement step

We compared our Off-BPG with the BPG in the policy improvement step. In our experiment, we examined two conditions of the sample size, $M = 5, 10$, which was used in each policy evaluation step. The policy evaluation with a small M can make large variance of the policy gradient estimate, and then the policy improvement can be slow and unstable. Fig.4 (a) shows the Euclidean distance between the optimal parameter and the estimated parameter: $||\boldsymbol{\theta}_{opt} - \boldsymbol{\theta}||$. Our Off-BPG with only 5 samples converged much faster than the BPG with 10 samples. On the other hand, the BPG with 5 samples could not converge but diverge. Fig.4 (b) shows that the parameter, which is the standard derivation of the policy, was updated to an incorrect direction by 5 sample size in the BPG, while Off-BPG worked toward a correct direction. This is because the policy gradient could not be estimated by that sample size passibly.



(a)



(b)

Figure 4: (a) Euclidian distance between the optimal and the estimated policy parameters. (b) Learning behavior of the policy parameter, which is the standard derivation of the policy.

6 Conclusion and Future Work

In this study, we proposed a reinforcement learning algorithm with active sampling based on Gaussian process (GP), called "Off-BPG". Our algorithm used the estimation variance of the policy gradient to explore the most efficient samples. Computer experiments showed that our algorithm could drastically reduce variance and realize fast convergence of the policy. However, the applicability has so far been shown in a simple bandit problem. Then, we will extend the applicability of our Off-BPG to more realistic problems as a future work.

References

[1] Ghavamzadeh, M., and Engel, Y. (2007). Bayesian policy gradient algorithms. Advances in Neural Information Processing Systems 19. Cambridge, MA: MIT Press.

[2] O'Hagan, A. (1991). Bayes-Hermite quadrature. Journal of Statistical Planning and Inference, 29, 245-260

[3] Williams and Rasmussen (1996). Gaussian processes for reguression. NIPS, Volume 8. The MIT Press.

[4] Baxter, J. and Bartlett, P.L. (2001) "Infinite-Horizon Policy-Gradient Estimation", Volume 15, pages 319-350.